UNITED STATES PATENT AND TRADEMARK OFFICE

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/943,563	08/30/2001	Maria-Jose Arbulu Barturen	FR920010023US1	2529

46590	7590	12/04/2006	

MYERS BIGEL SIBLEY SAJOVEC P.A.
PO BOX 37428
RALEIGH, NC 27627

EXAMINER
PHAM, CHRYSTINE

ART UNIT	PAPER NUMBER
2192	

DATE MAILED: 12/04/2006

Please find below and/or attached an Office communication concerning this application or proceeding.

BEFORE THE BOARD OF PATENT APPEALS
AND INTERFERENCES

Application Number: 09/943,563
Filing Date: **08/30/2001**
Appellant(s): BARTUREN ET AL.

MAILED

DEC 0 4 2006

Technology Center 2100

D. Randal Ayers
Registration No. 40,493
For Appellants

EXAMINER'S ANSWER

This is in response to the appeal brief filed 08/30/2006 appealing from the Office action

mailed 05/23/2006.

(1) Real Party in Interest

A statement identifying by name the real party in interest is contained in the brief.

(2) Related Appeals and Interferences

The examiner is not aware of any related appeals, interferences, or judicial proceedings which will directly affect or be directly affected by or have a bearing on the Board's decision in the pending appeal. .

(3) Status of Claims

The statement of the status of claims contained in the brief is correct.

(4) Status of Amendments After Final

The appellant's statement of the status of amendments after final rejection contained in the brief is incorrect. The claims of the present appeal are the claims presented in March **10**, 2005 Amendment (not March **8**, 2005).

(5) Summary of Claimed Subject Matter

The summary of claimed subject matter contained in the brief is correct.

(6) Grounds of Rejection to be Reviewed on Appeal

The appellant's statement of the grounds of rejection to be reviewed on appeal is substantially correct. The change (indicated in **bold**) is as follows:

1. The rejections of Claims 1-2, 4-8, 10, 12-15 and 17 under 35 U.S.C 102(e) as

anticipated by U.S. Patent No. 6,427,230 to Goiffon et al. ("Goiffon").

2. The rejections of Claims 3 and 9 under 35 U.S.C. 103(a) as **unpatentable**

over Goiffon in view of U.S. Patent No. 5,974,454 to Apfel et al. ("Apfel").

3. The rejections of Claims 11 and 16 under 35 U.S.C. 103(a) as **unpatentable**

over Goiffon in view of U.S. Patent No. 6,110,228 to Albright et al. ("Albright").

(7) Claims Appendix

A substantially correct copy of appealed claims 1-17 appears on the Appendix to

the appellant's brief. The minor errors are as follows: The correct status for claims 6,

12, 13-17 as of March 10, 2005 are (currently amended).

(8) Evidence Relied Upon

6,427,230	GOIFFON ET AL	6-2002
5,974,454	APFEL ET AL	10-1999
6,110,228	ALBRIGHT ET AL.	8-2000

(9) Grounds of Rejection

The following ground(s) of rejection are applicable to the appealed claims:

Claims 1-2, 4-8, 10, 12-15 and 17 are rejected under 35 U.S.C. 102(e) as

being anticipated by U.S.Patent No. 6,427,230 to Goiffon et al. ("Goiffon").

Claims 3 and 9 are rejected under 35 U.S.C 103(a) as unpatentable over

Goiffon in view of U.S. Patent No. 5,974,454 to Apfel et al. ("Apfel").

Claims 11 and 16 are rejected under 35 U.S.C 103(a) as unpatentable

over Goiffon in view of U.S. Patent No. 6,110,228 to Albright et al. ("Albright").

(10) Response to Argument

Argument III (Brief, pages 9-14)

□ **Argument A** (Brief, pages 11-14) (for claims **1, 4-6,** and **8**). Appellants divided

Argument A is into 3 sub-arguments (labeled herein as A1, A2 and A3). Each

sub-argument A1, A2 and A3 is followed by its corresponding response.

Argument A1 (Brief, pages 11-13) - "Goiffon does not disclose a system for

managing a process of delivery of software products". Essentially, Appellants

contend that since the "elements" that are stored in the Element Inventory

102 are only objects or metadata (i.e., data about data), the export operation,

thus, does not anticipate "a process of delivery of software products to target

software execution units".

❖ *Response to Argument A1*

As an initial matter, it should be noted that the first, second and third "sub-systems", as recited in at least claim 1 are interpreted as software means for "identifying ... software components ...", "creating ... software product package from identified components ... " and "distributing the created software product package ..." respectively. Furthermore, Mission-Specific Facilities 108 and Remote Object Management System 107 of FIG.1 and col.8:29-35 collectively disclose tools and processes that are required to **integrate** new applications (i.e., software components) with existing applications and/or **reimplement** all or parts of existing applications within (i.e., residing on) different systems and platforms (i.e., remote systems 107). As will be discussed further in details below, in order re-implement or integrate (i.e., combine or build a package from) software components that reside on different systems/platforms, the software components have to be retrieved from the different systems/platforms (i.e., distributed and delivered to and from the different systems/platforms) for integration and re-implementation purposes.

As acknowledged by Appellants (Brief, page 12, last paragraph), the **packages** created in Goiffon **contain actual software modules** and are migrated (i.e., **distributed and/or delivered**) to a new platform (i.e., remote system or **target software execution unit**).

Second, as has been pointed out in the Advisory Action (page 2, last 3 paragraphs) and Final Office Action (pages 3, 5-6), step 1828 of FIG.18B clearly discloses the **writing** (i.e., storing) the newly created package "element" (i.e., "element" of type *package*) to the Element Inventory 102 via the "create element" service call. It should be understood that since the "element" to be stored in the Element Inventory is of type *package*, it includes the **actual software code/modules** (as opposed to just objects or metadata describing the software code/modules) which make up the *package*. It should be understood also that the "element" of type *package* being written (i.e., stored) to the Element Inventory 102 <u>includes</u> a **combination of existing and re-usable packages** (i.e., "elements" of type *package*) and is explicitly taught by Goiffon in col.5:30-42 which reads, "... **existing package objects may be used during the creation of still other package objects**. That is, package objects may be created that model software **packages that include other software packages...**" (Emphasis added).

Third, as established in the Final Office Action (page 5), Goiffon discloses the export function on col.7:32-36 (see also FIG.2B & associated text) as followed: "The **export function** provides a **copy (i.e., delivery) of an element** to the **remote system** (i.e., Remote Object Management System 107 FIG.1 & associated text), whereas the **import function receives a copy of an element** from a remote system and **stores the copy within the Element Inventory 102**" (i.e., **centralized**) (Emphasis added). It should be

understood that the export function cannot exist without the import function. It

should be understood also that when an "element" of type *package* is being

exported (i.e., distributed and/or delivered) to a remote system, it contains the

actual software code/modules as well as "elements" (i.e., metadata)

describing the actual software code/modules. The exported *package*, which

contains the actual software code/modules, is then received by the import

function on the remote system and is copied (i.e., written) to the Element

Inventory 102 residing on the remote system. Since, the remote system can

create new packages based on *existing packages* <u>and</u> the **software**

code/module stored therein, it should be understood that the *existing*

packages include imported packages which are received by the import

function on the remote system, which are stored in the **centralized** Element

Inventory 102 as established above. Thus, contrary to Appellants' argument,

Goiffon clearly anticipates "a process of delivery of software products".

Argument A2 (Brief, pages 13-14) - "Goiffon does not disclose a central

repository for storing software components of at least one software product".

❖ *Response to Argument A2*

In col.8:48-55, Goiffon specifically states "After a group or data **components**

have been identified ... Element Packager 118 is utilized to **build the**

identified elements into a package that includes all of the code and data

necessary to transform the group of components. To perform this function,

the **Element Packager** must **extract** additional information about each of **the elements from Element Inventory 102**" (Emphasis added). Col.30:14-30 specifically states "**Elements of type 'Package' are created by the Element Packager 118** when the user selects a '**Create Package' Button 708** of FIG.7... The **newly-created element instance is written (i.e., stored) to the Element Inventory 102** using the 'Create Element' service call" (Emphasis added). Furthermore, as discussed above in *Response to Argument A1*, the Element Inventory 102 contains packages (with actual software code/modules stored therein) that are imported from a remote system. Also as discussed above, all imported and re-usable packages and re-usable software code/modules residing in Element Inventory 102 can be reused to create new packages. Thus, contrary to Appellants' argument, Element Inventory 102 clearly anticipates a **central** repository for storing "reusable" software components (i.e., packages or software code/modules) of at least one software product (i.e., package).

Argument A3 (Brief, page 14) - "Goiffon does not disclose a third sub-system for distributing the at least one software product".

❖ *Response to Argument A3*

Col.8:58-67 of Goiffon explicitly teaches wrapping the created Package with layers of software called "wrapper" that provide an interface bridge between the wrapped elements that makes that package accessible from a different

operating environment (i.e., remote system 107). It is clear that the wrapping

the created package anticipates preparing the package for distribution to

different remote system 107. Furthermore, as discussed above in *Response*

to Argument A1, Goiffon's exporting packages, which contain actual software

code/modules) to remote system and storing the imported packages (and the

actual software code/modules contained therein) in the Element Inventory

102 of the remote system to facilitate the creation of other new packages

clearly anticipates distributing the at least one software product.

▫ **Argument B** (Brief, page 15)(for claims 2 and 10) – "Goiffon does not teach a

central distribution repository for storing at least one software product package".

❖ *Response to Argument B*

As discussed above in *Response to Argument A1*, the Element Inventory 102

contains packages (with actual software code/modules stored therein) that

are imported from the remote system. Also as discussed above, all imported

and re-usable packages and re-usable software code/modules stored in

Element Inventory 102 can be reused to create new packages. Thus,

contrary to Appellants' argument, Element Inventory 102 clearly anticipates a

"central distribution repository for storing at least one software product

package" (recited in claims 2 and 10) created by the second-subsystem (i.e.,

sub-step or sub-process) (see step 1828 of FIG.18B of Goiffon) from the

identified software components.

❏ **Argument C** (Brief, pages 15-16)(for claim 7) - "Goiffon does not teach a sixth subsystem for recording information provided by at least one of the first through fifth subsystem ... during delivery of the software product".

❖ *Response to Argument C*

Appellants acknowledge that the Final Office Action cites to line 227 of FIG.2A and line 240 of FIG.2B of Goiffon. Appellants further acknowledge that line 227 connects Client Server 216 (i.e., remote system or target software execution unit) to AIM Server 214. As clearly disclosed in FIG.2B of Goiffon, line 227 is directly connected to Import/Export Files via **line 240**. In .col.14:20-33, Goiffon explicitly discloses that the **Export Elements** service reads elements from Element Inventory 102 and **writes** them into a file (i.e., record information) as is indicated by dashed line 240. Similarly, the Import Elements service reads elements from a file (received from a remote system) and writes them (i.e., record information) into the Element Inventory 102 as indicated by dashed line 240. It is clear that line 240 anticipates recording information provided by at least one of the first through fifth subsystem ... during delivery (i.e., exporting/importing) of the software product (i.e., package).

    ❑ **Argument D** (Brief, pages 16-19)(for claim 12-15 and 17) – Appellants refer to

previous arguments to assert patentability of claim 12 and depending claims 13-

15 and 17.

    ❖ *Response to Argument D*

Since claim 12 recites limitations already addressed above in claims 1-2,

Appellants are referred to the above responses Arguments A and B.

Argument IV (Brief, page 19)(for claims 3 and 9) – "Goiffon and Apfel have nothing

in common, and are directed to different problems and solutions. There's no

motivation to combine the two disclosures".

    ❖ *Response to Argument IV*

As has been established in the Final Office Action (page 11), col.6:49-col.7:10

(text associated with computer 20 of FIG.1) of Apfel explicitly discloses

computer 20 (i.e., target software product execution unit) sending a query 100

to server 80a requesting a upgrade package (i.e., **packaged software**

product), to be installed (i.e., distributed to) on computer 20. As has been

established in the Final Office Action (page 11), col.8:52-col.9:5 of Apfel

explicitly states that "the query [100] is structured to **identify the information**

needed to determine which **upgrade package** UR is required by computer

20. This **information** may include the **versions of the program module**

components to be upgraded...". It is clear from this passage that the

upgrade package to be installed on computer 20 <u>includes</u> **multiple** program

module (i.e., **software) components,** which have been identified in the

query. Thus, contrary to Appellants' argument, Goiffon and Apfel are clearly

analogous art because they are clearly directed to distributing software

products (i.e., software packages) to end users (i.e., target software product

execution units). As such, it would have been obvious to one of ordinary skill

in the pertinent art at the time the invention was made to incorporate the

teaching of Apfel into that of Goiffon for the inclusion of assigning (i.e.,

associating) a role (i.e., operating system) for each software package and

distributing said package to target execution units belonging to an

environment according to (i.e., matching) said role (as cited in claim 3). And

the motivation for doing so, as established in the Final Office Action (page 12)

would have been to provide software packages to different execution units

that are operating on different operating systems (see Apfel col.9:30-45).

Argument V (Brief, page 20)(for claims 11 and 16) – "Albright is not cited as

disclosing any of the recitations from claims 8 and 12".

   ❖ *Response to Argument V*

It should be noted that Albright was not relied upon to teach recitations of claims

8 and 12. As clearly established in the Final Office Action (pages 12-14), Albright was

cited as disclosing the limitation "building source code components of the software

product and storing the result of building in the central repository" (recited in claim 11)

and "wherein the built software product comprises execution code that is generated

from a source code component stored in the central repository" (recited in claim 16).

(11) Related Proceeding(s) Appendix

No decision rendered by a court or the Board is identified by the examiner in the

Related Appeals and Interferences section of this examiner's answer.

(12) Copy of Claim Rejections (regenerated herein for completeness)

Claim Rejections - 35 USC § 102

1. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that
    form the basis for the rejections under this section made in this Office action:
    A person shall be entitled to a patent unless –
    (e) the invention was described in (1) an application for patent, published under
    section 122(b), by another filed in the United States before the invention by the
    applicant for patent or (2) a patent granted on an application for patent by
    another filed in the United States before the invention by the applicant for patent,
    except that an international application filed under the treaty defined in section
    351(a) shall have the effects for purposes of this subsection of an application
    filed in the United States only if the international application designated the
    United States and was published under Article 21(2) of such treaty in the English
    language.

2. Claims 1, 2, 4-8, 10, 12-15, and 17 are rejected under 35 U.S.C. 102(e) as being
    anticipated by Goiffon et al. (US 6,427,230 B1; hereinafter *Goiffon*).

Claim 1
    Goiffon teach an integrated data processing system (see at least *object
management system 100* Fig.1 & associated text) for managing a process (i.e., <u>method</u>)
of delivery of software products (see at least *object repository, software constructs,
packages* Abstract; *export function, element, remote system* col.7:23-40; *Export
elements, Client Server 216* col.14:20-25; *240* Fig.2B & associated text; *227* Fig.2A &
associated text) to target software product execution units in a network environment
(see at least *client server 216* Fig.2A & associated text; col.2:53-56; col.3:20-32;
col.4:15-67; col.8:47-57; FIG.1 & associated text), comprising:
    o a central repository for storing software components at least one software
        product (see at least *object repository, software constructs, packages* Abstract;

AIM Server 214, Element Repository 220 Fig.2B & associated text; col.12:7-15; col.12:23-67; *Host A 228, Memory 229* Fig.2B & associated text; *Host A 228, Memory 229, data modules* col.12:57-col.13:20);

o a first sub-system for identifying within the central repository software components (see at least *selecting, data modules* col.2:53-56; *users selectively include, data modules* col.3:1-45) of software product (see at least *integrated package* col.2:53-56; *creation, data packages, data modules* col.3:20-45) be delivered (see at least *1808, 1816* Fig.18A & associated text; col.23:1-10);

o a second sub-system for creating at least one software product package (see at least *package creation, package definition* col.4:14-67) from the identified software components identified by the first sub-system (see at least *software constructs, user interface* col.4:14-67; *Element Packager 118* Fig.1 & associated text), and

o a third sub-system for distributing the least one software product package created by the second sub-system to the target software product execution units and installing the software product package thereon (see at least *export function, element, remote system* col.7:23-40; *Export elements, Client Server 216* col.14:20-25; *240* Fig.2B & associated text; *227* Fig.2A & associated text; col.2:53-56; col.3:20-32; col.4:15-67; col.8:47-57; FIG.1 & associated text).

Claim 2
The rejection of base claim 1 is incorporated. *Goiffon* further teach a software package distribution repository for storing the at least one software product package created by the second sub-system from the identified software components (see at least *1024* Fig.10 & associated text; *1808, 1816* Fig.18A & associated text; *1828* Fig.18B & associated text; *Create Elements, Update element* col.14:54-63).

Claim 4
The rejection of base claim 1 is incorporated. *Goiffon* further teaches first sub-system manages storage in the central repository of the software components software product to be delivered (see at least *object repository, software constructs, packages* Abstract).

Claim 5
The rejection of base claim 1 is incorporated. *Goiffon* further teach a fourth sub-system for performing a building process of software code components among the identified software components of the software product be delivered (see at least *interdependencies, group modules, packages* col.2:22-47; *package creation, software constructs, interdependencies* col.4:14-67; *data components, Element Packager 118, build, identified elements* col.8:47-67), the fourth sub-system storing result of building process in the central repository (see at least *object repository, software constructs, packages* Abstract).

Claim 6

The rejection of base claim 1 is incorporated. *Goiffon* further teach a fifth sub-system managing a process applying changes (i.e., new version) to at least one software product distributed by the third sub-system (see at least *package objects, forming relationships, interdependencies* col.5:30-40; *package, renovation operation* col.4:15-25; *data modules, functional unit, renovation operations* col.3:39-45; *renovation, tools, new versions* col.8:1-17).

Claim 7
The rejection of base claim 1 is incorporated. *Goiffon* further teach a sixth sub-system for recording information provided by at least one of the first through fifth sub-systems the integrated data processing system during delivery of the software product (see at least *227* Fig.2A & associated text; *240* Fig.2B & associated text).

Claim 8
Claim recites a method for delivering software products to target software product execution units in a network environment as have been addressed in claim 1, therefore, is rejected for the same reasons as cited in claim 1.

Claim 10
The rejection of base claim 8 is incorporated. Claim recites limitations, which have been addressed in claim 2, therefore, is rejected for the same reasons as cited in claim 2.

Claim 12
Goiffon teach a method of developing and installing a software product on a plurality of target computers (see at least *object repository, software constructs, packages* Abstract; *export function, element, remote system* col.7:23-40; *Export elements, Client Server 216* col.14:20-25; *240* Fig.2B & associated text; *227* Fig.2A & associated text), the method comprising:
- o storing a plurality of components in a central repository (see "central repository" claim 1);
- o using at least some of the plurality of stored components to build the software product (see "second sub-system" claim 1);
- o storing the built software product in the central repository (see "software product package distribution repository" claim 2)
- o creating an installable software package that includes at least some of the plurality of components and the built software product (see "third sub-system" claim 1);
- o storing the installable software package in a second repository;
- o distributing the installable software package to at least some of the plurality of target computers (see "target software product execution units", "third sub-system" claim 1);

o   installing the distributed installable software package on the at least some of the
    plurality of target computers (see "target software product execution units", "third
    sub-system" claim 1);

Claim 13

The rejection of base claim 12 is incorporated. *Goiffon* further teach wherein ·
software product comprises a newly developed software product (see at least *package
objects, forming relationships, interdependencies* col.5:30-40; *package, renovation
operation* col.4:15-25; *data modules, functional unit, renovation operations* col.3:39-45;
renovation, tools, new versions col.8:1-17).

Claim 14

The rejection of base claim 12 is incorporated. Claim recites limitations, which
have been addressed in claim 6, therefore, is rejected for the same reasons as cited in
claim 6.

Claim 15

The rejection of base claim 12 is incorporated. *Goiffon* further teach recording
information regarding the software product in a tracking sub-system (see at least *object
repository, software constructs, packages* Abstract).

Claim 17

The rejection of base claim 12 is incorporated. *Goiffon* further teach providing a
configuration management subsystem that controls and manages different versions of
the software components stored in the central repository (see at least *groups of re-
usable code, data modules, other groups* col.3:15-20; *updated versions, associated
elements, interrelated, Element Inventory* col.8:10-17).

Claim Rejections - 35 USC § 103

3.   The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all
obviousness rejections set forth in this Office action:

*(a) A patent may not be obtained though the invention is not identically disclosed
or described as set forth in section 102 of this title, if the differences between the
subject matter sought to be patented and the prior art are such that the subject
matter as a whole would have been obvious at the time the invention was made
to a person having ordinary skill in the art to which said subject matter pertains.
Patentability shall not be negatived by the manner in which the invention was   ·
made.*

4.   Claims 3 and 9 are rejected under 35 U.S.C. 103(a) as being unpatentable over
Goiffon in view of Apfel et al. of record (US 5974454, hereinafter *Apfel*).

Claim 3

The rejection of base claim 1 is incorporated. *Goiffon* further discloses the third sub-system distributes the at least one software product package to target software product execution units belonging to at least one environment (see at least *different operating environment* col.8:58-67). *Goiffon* does not expressly disclose said environment according (i.e., matching) to at least one role assigned to the at least one software product package. However, *Apfel* teaches assigning (i.e., associating) each software product package with a role (i.e., environment or operating system) and distributing said package to target software product execution units belonging to an environment according (i.e., matching) said role (e.g., see *configuration of computer 20, different upgrade package* col.6:65-67; *type of operating system, upgrade package URL, query* col.8:52-9:5; col.9:35-42). *Goiffon* and *Apfel* are analogous art because they are directed to distributing software packages. It would have been obvious to one of ordinary skill in the art at the time of the invention to incorporate the teaching of *Apfel* into that of *Goiffon* for the inclusion of assigning (i.e., associating) a role (i.e., operating system) for each software package and distributing said package to target execution units belonging to an environment according (i.e., matching) said role. And the motivation for doing so would have been to provide software packages to a variety of execution units belonging to different operating systems (see at least *Apfel* col.9:30-45).

Claim 9
The rejection of base claim 8 is incorporated. Claim recites limitations, which have been addressed in claim 3, therefore, is rejected for the same reasons as cited in claim 3.

5.   Claims 11 and 16 are rejected under 35 U.S.C. 103(a) as being unpatentable over *Goiffon* in view of Albright et al. of record (US 6110228, hereinafter *Albright*).

Claim 11
The rejection of base claim 10 is incorporated. *Goiffon* does not expressly disclose building source code components and storing the result of building in the central repository. However, *Albright* discloses a system and method of distributing executable code to client computers, wherein the source code components are built and the result is store in the central repository (see at least *service site, executable code, source code* Abstract). *Goiffon* and *Albright* are analogous art because they are both directed to distributing and installing software in target computers. It would have been obvious to one of ordinary skill in the pertinent art at the time the invention was made to incorporate the teaching of *Albright* into that of *Goiffon* for the inclusion of generating/building execution code from source code components stored in central repository. And the motivation for doing so would have been to eliminate the necessity of including within the software upgrade (to be distributed and installed in target computers) many lines of [source] code that are dedicated only to permitting the target computers to add software upgrade, thus improving the efficiency of the upgrading process (see *Albright* col.3:1-30).

Claim 16

The rejection of base claim 12 is incorporated. *Goiffon* does not expressly disclose wherein the built software product comprises execution code that is generated from a source code component stored in the central repository. However, *Albright* discloses a method of installing updated executable code (i.e., built software product) in target computers wherein the executable code is generated from a source code component stored in the central repository (see *central software service site, customer, remote location, updated executable code, installing fixes, source code, program product* in Abstract). *Goiffon* and *Albright* are analogous art because they are both directed to distributing and installing software in target computers. It would have been obvious to one of ordinary skill in the pertinent art at the time the invention was made to incorporate the teaching of *Albright* into that of *Goiffon* for the inclusion of generating execution code from source code stored in central repository. And the motivation for doing so would have been the same as has been cited for claim 11.

For the above reasons, it is believed that the rejections should be sustained.

Respectfully submitted,

Chrystine Pham *CP*

Conferees:

Tuan Dam, SPE AU2192

TUAN DAM
SUPERVISORY PATENT EXAMINER

Eddie Lee, SPE 2100